PHOTONICS WORKSHOP

CONSTRUCTING A DOLPI CAMERA
MAKING POLARISATION VISIBLE

ViNN:Lab Think:Make:Share
Technische Hochschule Wildau Technical University of Applied Sciences
PHABLABS 4.0

“Constructing a Dolpi camera” is a workshop that can be executed in Fab Labs.

Properties of this workshop

Target audience:

Young professionals (18+ years old)

Time planning (130 min):

Understanding polarization and the working principle	25 min
Constructing the casing	25 min
Soldering	20 min
Programming the Raspberry Pi	20 min
Final assembling	20 min
Testing	20 min

Estimated costs: € 30

Content

Summary

In this project we'll make a Dolpi camera, that is, a camera that takes pictures based not only on the intensity and wavelength of light (like normal cameras do), but also based on its polarization.

To achieve this, we'll wire an ordinary cheap webcam and a servomotor with a Raspberry Pi Zero W. The servomotor has a polarizer attached to it. So when the servomotor moves, it rotates the polarizer. As the polarizer is in front of the camera lens, we get different pictures for each rotation angle. By comparing the pictures (the software does all this for us), we get a polarization resolved picture.

The camera will be accessible with any device that has WiFi access, for example a laptop or a smartphone, via an intuitive web interface. As an application, this camera might theoretically be used to detect landmines in the future.

Project goals

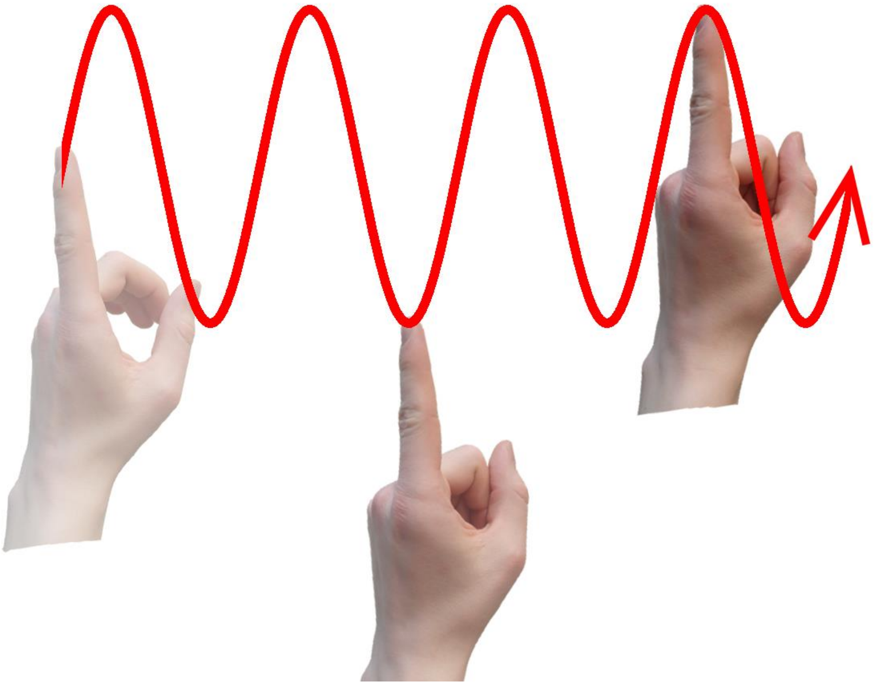
The participants

- learn about the wave nature of light, especially about the phenomenon of polarization
- get into contact with the modern field of laser material processing
- use their manual skills in order to construct a box made of plywood
- learn how to solder pin heads on a circuit board
- get to learn the operating system Linux on a Raspberry Pi
- can take a finished product home

What is polarisation?

Light is composed of electromagnetic waves that propagate in space.

Try to trace the path of a single light wave with your hand. You will find that there are different directions in which the wave can oscillate. It might go up and down or it might go back and forth – or maybe something in between:

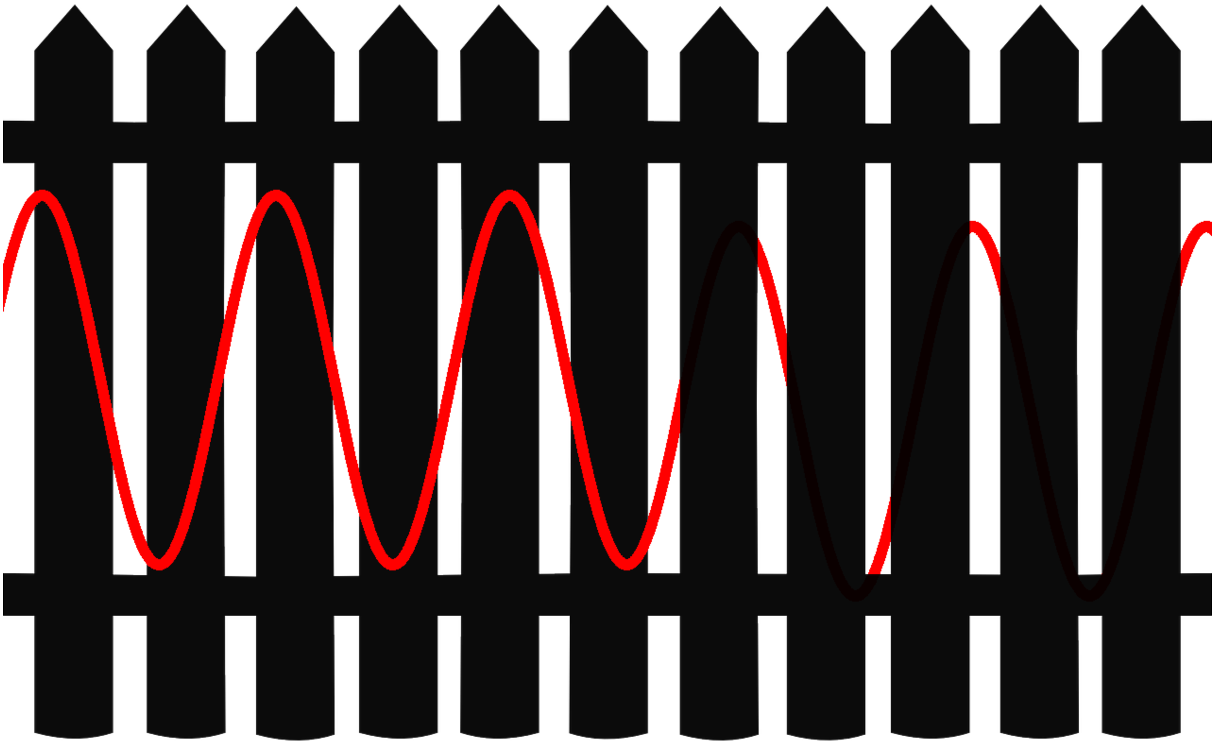


This phenomenon is called **polarization**. The plane in which a wave oscillates is called a **plane of polarization**.

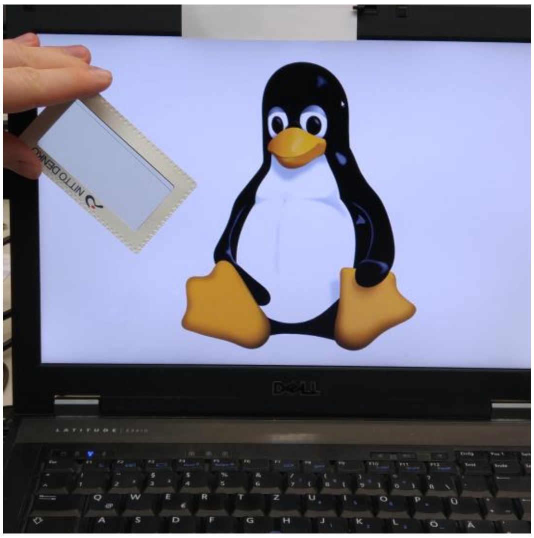
If all waves coming from a light source share the same plane of polarization (that is, all oscillate in the same direction) the light is said to be **linearly polarized**. Typical gadgets that emit linearly polarized lights are, for example, lasers and LCD displays.

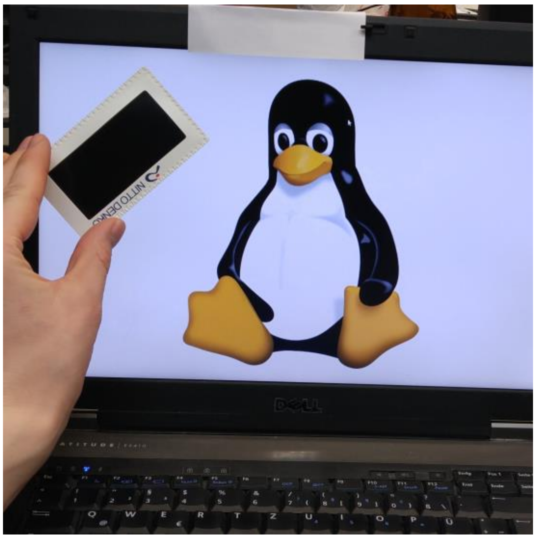
But most light we deal with in our daily life is not linearly polarized – instead, all the individual light waves have a different plane of polarization. In this case we call the light **unpolarized**.

Using special **polarizers** we can filter light according to its polarization. A polarizer is an object that only lets through light waves with a certain plane of polarization – all the other waves are blocked. This way, you can compare it with a picket fence which would only let through items oscillating in the direction of the pickets:



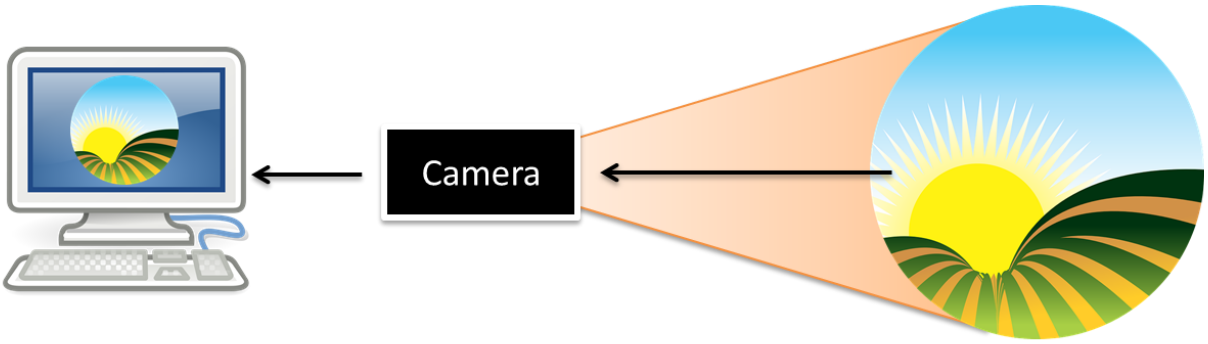
You can try this out with a polarizer and an LCD display (from a laptop, a TV set, a smartphone …). Just hold the polarizer in front of the display and rotate it. As the LCD emits linearly polarized light, the light is only let through when the orientation of the polarizer and the plane of polarization match. If you turn the polarizer by 90°, you‘ll notice that all the light is blocked:





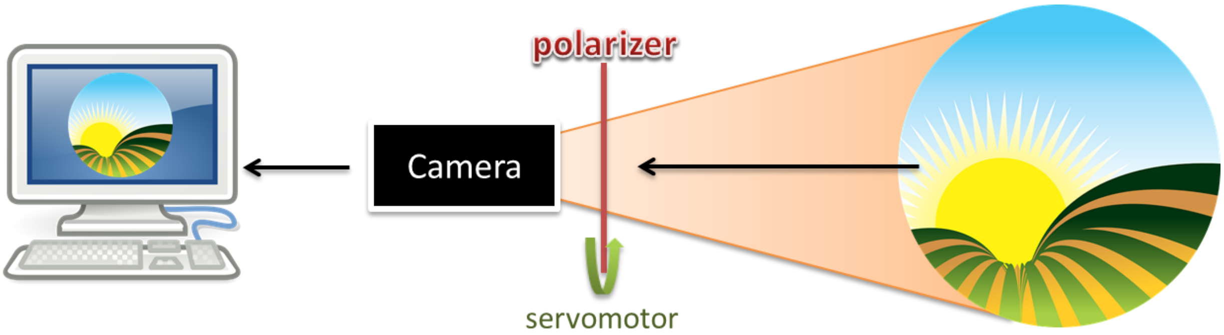
Working principle

Let's examine how "normal" digital cameras take pictures. The camera takes the light from its surroundings using an optical system made up of lenses and projects it onto a sensor (like a CCD or CMOS). The light carries information about its **wavelength** (defining the colour) and **intensity** (defining the brightness) and if you combine both, you'll get a nice picture like the following:



To recap: Cameras (just like our human eyes) take into account two properties of light: wavelength and intensity. But wait: Didn't we learn in the last section that light – as a transversal wave – also shows the phenomenon called polarization?!

So we might in fact add another physical quantity to the image – the **degree of polarization**. And this is what our Dolpi camera will do. Here is how:



We'll simply put a polarizer in front of the camera. As you've learned, the polarizer will only let through light with a certain polarization. If you rotate the polarizer, you also rotate the plane of polarization of the portion of light that can go through.

When light is unpolarized, there should be no difference in the amount of light that can pass through when you rotate the polarizer. That is because all polarization angles are uniformly distributed. Thus, the camera picture will be quite the same.

However, if our light shows some linear polarization, there should be a difference. Like with the polarizer in front of the LCD screen (see last page) the amount of transmitted light will greatly vary for different rotation angles and therefore you get a different image.

Now, remember that we want to measure the degree of polarization to construct a picture out of it. The strategy might be as follows: We'll put the polarizer in front of our camera and take a picture. Next, we'll rotate the polarizer by 90° and take another picture at the same scene. Then, we take all the pixel intensity values $P_1(x,y)$ from picture one and all the pixel intensity values $P_2(x,y)$ from the second picture and calculate the absolute value of the difference $P(x,y)$:

$$P(x,y) = | P_1(x,y) - P_2(x,y) |$$

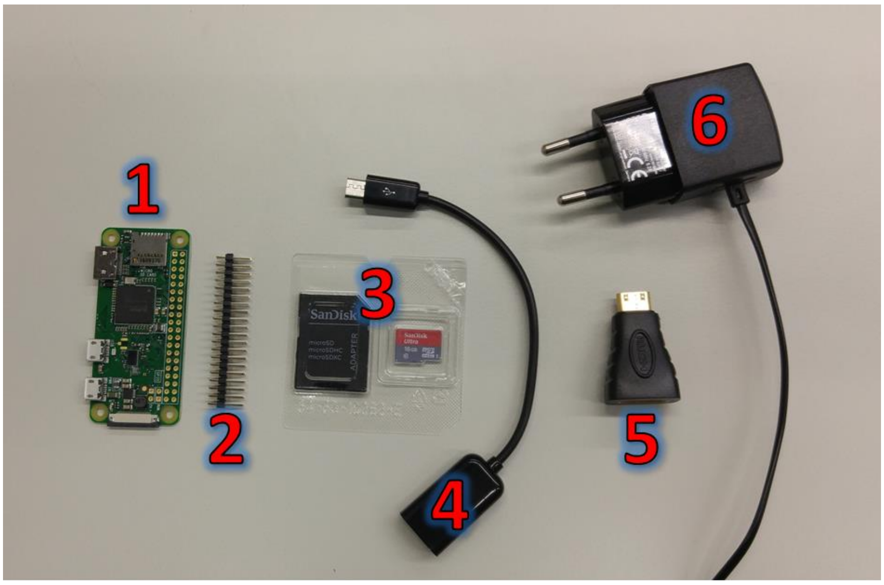
If the light is unpolarized, there should be no big difference between the two corresponding pixel intensities. Because of this, $P(x,y)$ should be close to zero. However, if the light is linearly polarized, there should be some difference and $P(x,y)$ should be substantially bigger than zero. If we now visualize P(x,y) as a picture, we'll clearly see the spots of the photographed scenery that emit or reflect linearly polarized light.

Note that we will elaborate our concept further by taking more than just two pictures and thus getting more reliable and precise data about the degree of polarization.

The rotation of the polarizer will be performed by a cheap servomotor. The camera will be an easy webcam and everything will be plugged to a Raspberry Pi.

What we need

For the electronics part, we take a **Raspberry Pi Zero W (1)**. This is a very small computer that is powerful enough to perform calculations with pictures (an Arduino would probably be too weak for this). In order to keep things short and simple, we will call it **RasPi** from now on. As we need to control a servomotor, we also need a **male pin header (2)** which will later be soldered to the RasPi. A **microSD card (3)** and an appropriate adapter or a card reader will be used to program the RasPi. To connect our webcam and a monitor, we also need a **USB-microUSB adapter (4)** and a **HDMI-microHDMI adapter (5)**. A **power supply (6)** (5 V, 1 A) is, of course, also necessary. Furthermore we will use a **USB hub** and an ordinary **PC**.



When it comes to the peripherals, we need a simple **USB webcam** and a **servomotor**. The webcam should be compatible with Linux as this is the operating system that is installed on our RasPi.

For the housing, we'll use common **plywood sheet** with a thickness of 3 mm. The different parts will be assembled using **wood glue**. And last but not least, we need a **polarizer in the form of a polarizing film sheet**!

Working procedure

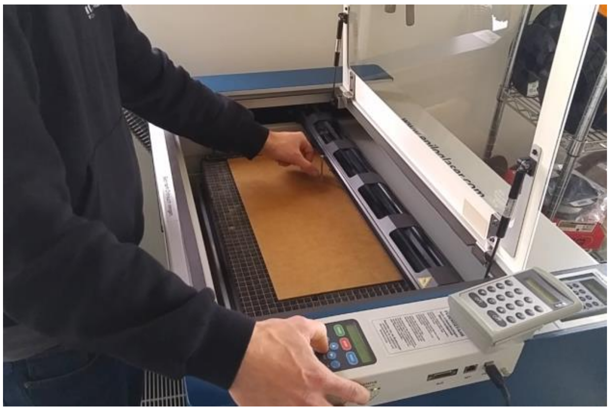
Step 1: Build the housing

In this step we'll be using a laser cutter machine. Make sure you've understood all necessary security regulations. Especially, it's important to keep the laser casing shut during operation!

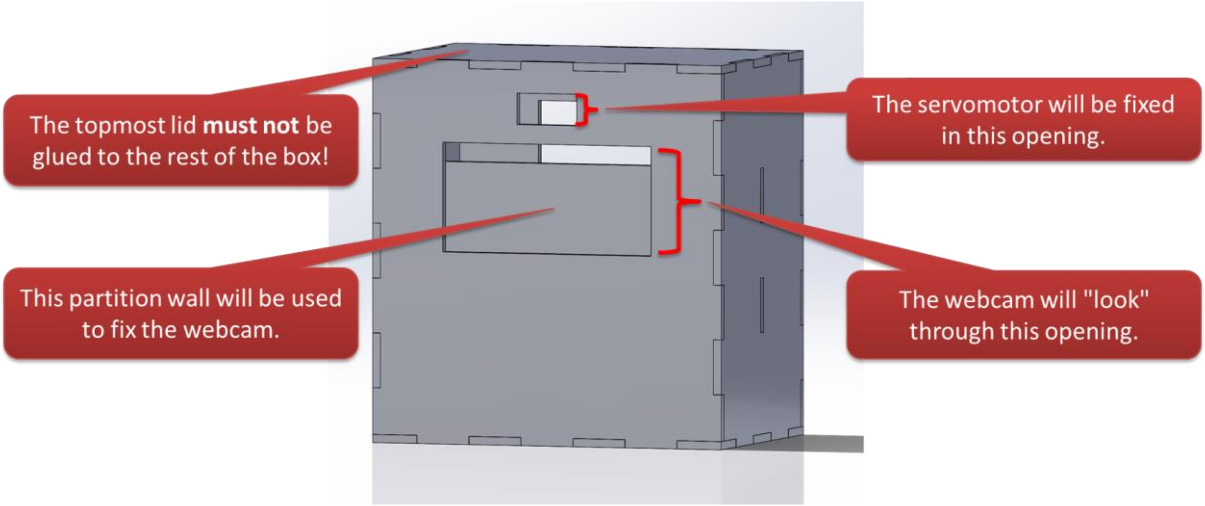
Take the plywood sheet and put it into the laser cutting machine. We provide a file with the designed pattern: `DolpiCam_housing.pdf`. Make sure you have the appropriate laser parameters set! For plywood with a thickness of 3 mm processed by a 40 W CO_2 laser we determined the following parameters to be optimal:

- Speed: 65 %
- Power: 90 %
- Pulse frequency: 500 Hz

Adjust the sheet within the laser cutter and let the process start:



Now, the laser head will cut out the parts from the plywood sheet. Once finished, take the plywood out of the machine and try to assemble the different parts, at first without using glue. It should look like this:

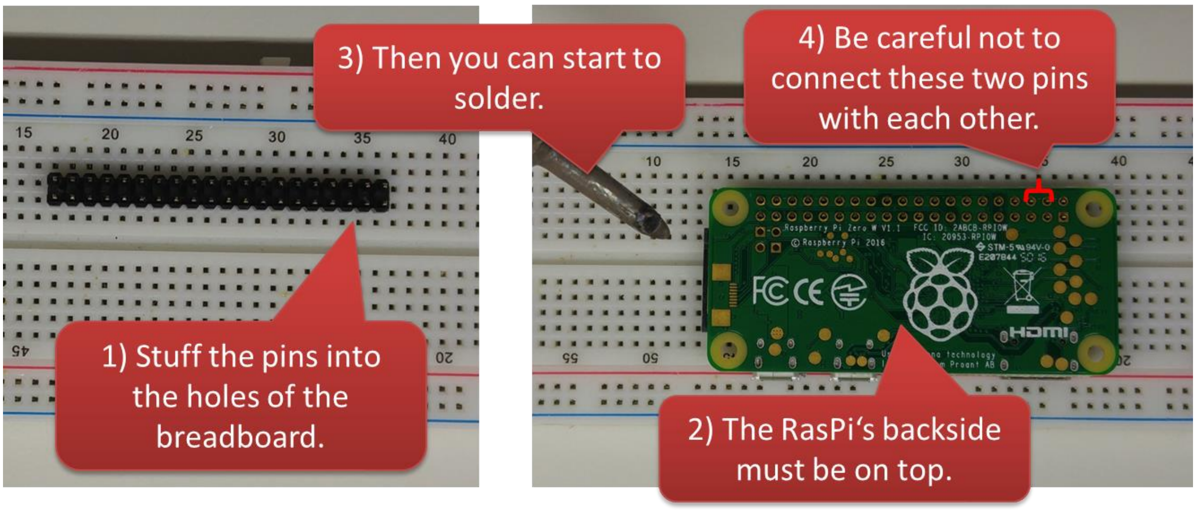

The topmost lid **must not** be glued to the rest of the box!
This partition wall will be used to fix the webcam.
The servomotor will be fixed in this opening.
The webcam will "look" through this opening.

Next, take some wood glue and glue the different parts together. However, it is important **not to glue the topmost lid** as we will have to include the electronics later on. The drying of the glue takes some time. Meanwhile you can proceed to the next steps.

Step 2: Solder the pin heads on the RasPi board

Other than the “full-grown” version of the Raspberry Pi, our cheap Raspberry Pi Zero comes without pins attached to it. This is why we have to solder some pins to the RasPi in order to later connect it to a servomotor (or perhaps other peripherals that do not use USB). Please be careful:

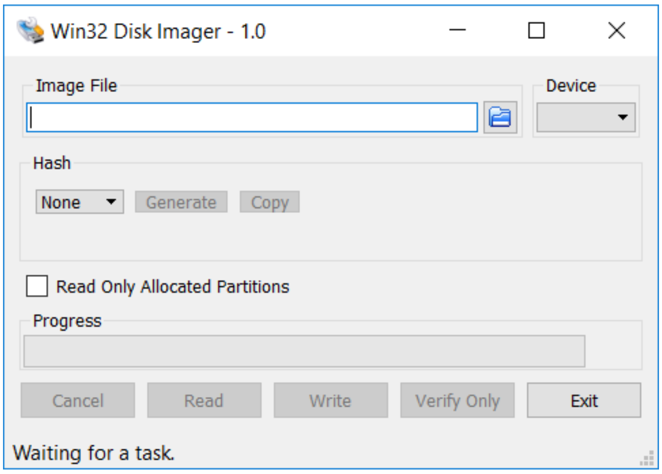
- The soldering iron will become very hot. It should never touch your skin.
- Do not apply heat on the RasPi for too long (not more than about two seconds per application) as this might otherwise damage the RasPi.


3) Then you can start to solder.
4) Be careful not to connect these two pins with each other.
1) Stuff the pins into the holes of the breadboard.
2) The RasPi's backside must be on top.

First, put the pin heads on a solderless breadboard and fix the RasPi on top of them, so that the backside of the board is on top (see figure). Next, start to solder the pins to the Pi. Be careful not to create any unwanted electrical connections between the different pins. Especially it is dangerous to link GND and VIN. This would result in a short circuit and might damage or even destroy the RasPi. If unsure, check the connection between two pins with a multimeter. The resistance between these two pins should at least be in the order of magnitude of 1 MΩ or so. (Use the multimeter after removing the RasPi with the pins from the breadboard as the electrical connections on the breadboard might otherwise affect the measured resistance.)

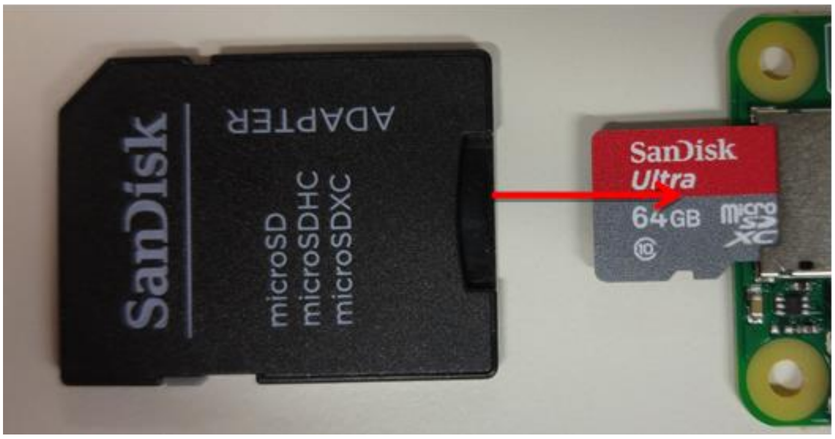
Step 3: Program the RasPi

The RasPi gets its information from a microSD card that contains the operating system and the other programs it has to execute. Thus, we have to write all this data on the microSD card. Stuff the card into an adapter and connect it to a PC. (If your PC already has an appropriate slot you can connect it directly, of course.)



Waiting for a task.

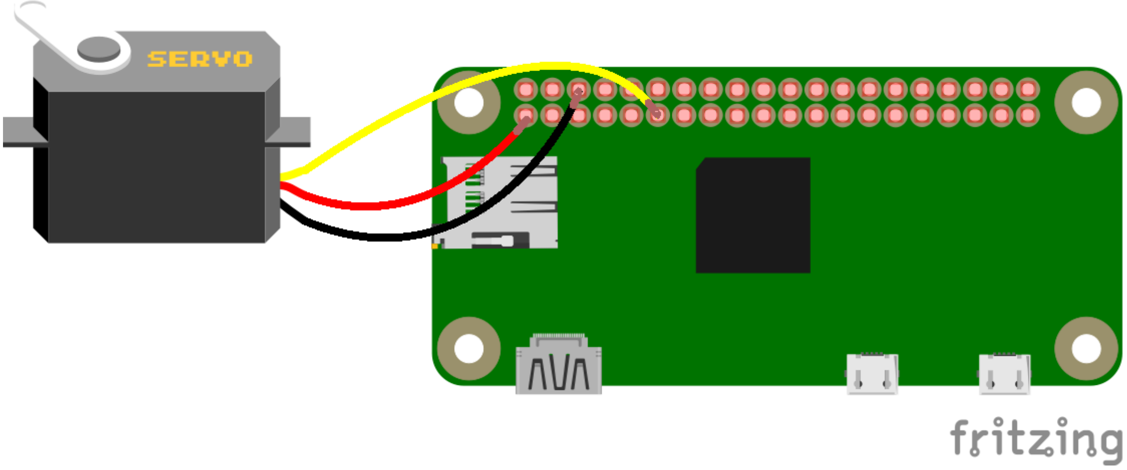
Luckily for you, we have already prepared an ISO image for you which you can simply write on the card in one step without worrying about all the different files that it contains. There are many programs that can do this. Here, we'll use Win32 Disk Imager. Start the program and choose the provided image file `DolpiCam.iso`, select the correct device containing the microSD card and click “Write”. Be very careful not to select the wrong storage medium or else there is a risk of data loss! Once the writing process has finished, remove the microSD from the PC/adapter and stuff it into the RasPi.



Later on, we'll take a closer look on what we've written on the microSD card, but for the moment, we don't care too much about this.

Step 4: Connect the Pi and see if it works

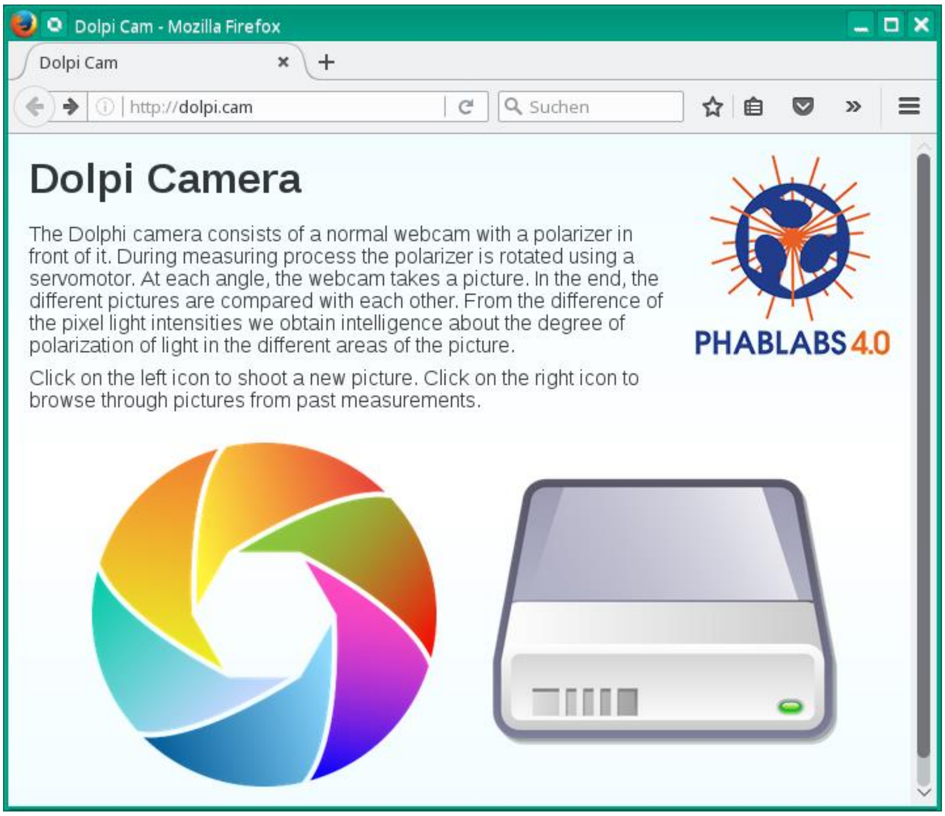
- Connect a USB hub to the microUSB-USB adapter and attach the latter to the Pi.
- Plug a computer mouse, a keyboard and the webcam to the hub.
- In order to see something, connect the miniHDMI-HDMI adapter to the Pi and link it to a monitor.
- Connect the black wire of the servomotor to pin 6 (GND), the red wire to pin 1 (3,3V PWR) and the yellow wire to pin 11 (GPIO 17) of the Pi. Use jumper wires if necessary.



- Connect the RasPi to the power supply. The Pi will now boot the operating system on its microSD card.

After about one minute, the desktop interface shows up on the monitor. You should also be able to use mouse and keyboard.

Now let's see if the Dolpi functionality is working as desired. Open the web browser Chromium and go to http://localhost/. A website like the following should show up:



Dolpi Camera

The Dolphi camera consists of a normal webcam with a polarizer in front of it. During measuring process the polarizer is rotated using a servomotor. At each angle, the webcam takes a picture. In the end, the different pictures are compared with each other. From the difference of the pixel light intensities we obtain intelligence about the degree of polarization of light in the different areas of the picture.

Click on the left icon to shoot a new picture. Click on the right icon to browse through pictures from past measurements.

Click on the colourful shutter on the left-hand side of the browser window (only once). The servomotor should now begin to move and the webcam should take pictures. After the measuring process has finished, you should see the resulting photos in the browser.

Now, take another PC with WiFi access (a smartphone is also OK), activate the wireless mode and see if you can find a network called "*DolpiCam_XYZ*", where *XYZ* is a random string. If yes, connect to the network, open a web browser and go to any website. You should again see the same interface and you should be able to start the measuring process just the same.

Troubleshooting:

- The camera doesn't work: Open up the application "Cheese". It should instantaneously show the signal from the webcam. If it doesn't, try to re-connect the camera and/or to reboot the Pi. If all this does not work, try the camera on another computer if it works at all. Keep in mind that we use Linux as operating system and that not all webcams are supported by Linux.

- The servomotor does not respond: Are all wires connected correctly? Connect a multimeter between pins 1 and 6 to see if there is really a voltage supply of 3.3 V.

- The WiFi network is not there: Move closer to the Pi.

Step 5: See *why* it works (optional)

In this step we’ll look at one of the program files that make our Dolpi cam run as desired. This step is recommended for those interested in computer science in general and programming in particular. Everyone else can move on to the next step.

In the RasPi user interface, open a file manager (that’s the folder icon in the control bar at the top of the screen) and browse to the directory `/home/pi/DolpiCam/`. Double-click on the file `server.py` to open it in a text editor. This is the script file that organizes the webserver that provides access to the Dolpi cam. It is written in the programming language *Python.* Scroll to the bottom of the file and identify the function `measure()`:

```
@app.route(‘measure/’)
def measure():
  "Start the measuring process by controlling the servo and the
webcam."
  ServoOn()

  # Determine the new measurement ID
  ID = NextID()

  # Shoot pictures at different angles
  img1 = ShootAtAngle(ID, 45)
  img2 = ShootAtAngle(ID, 90)
  img3 = ShootAtAngle(ID, 135)

  # Calculate the differences between the pictures
  diff1 = cv2.absdiff(img1, img2)
  diff2 = cv2.absdiff(img1, img3)
  diff3 = cv2.absdiff(img2, img3)

  # Calculate the maximum difference for each pixel and save the
resulting image
  maxdiff = np.fmax(np.fmax(diff1, diff2), diff3)
  cv2.imwrite(FilePath(ID, "diff.jpg"), maxdiff)
  ServoOff()
```

`measure()` is called each time the user clicks on the button to start the measurement process, that is, each time the URL `measure/` is called in a browser. First, the servomotor is switched on by calling the function `ServoOn()`. After this, we move the servomotor to three different angles (45°, 90° and 135°) and take a picture at each position. These pictures are subsequently stored in the variables `img1`, `img2` and `img3`.

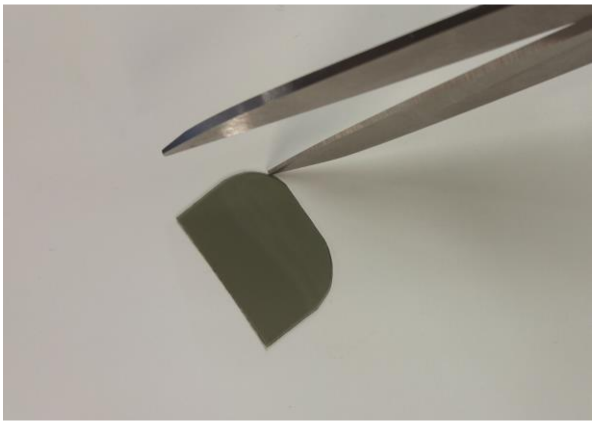
These three pictures represent the spatial light intensities for three different planes of polarization. For unpolarized light, there will be no big difference between these intensities (see the section about the working principle on one of the previous pages). However, when light is linearly polarized, there will be a difference. This is why we calculate the difference using the function `cv2.absdiff` for all three picture pairs.

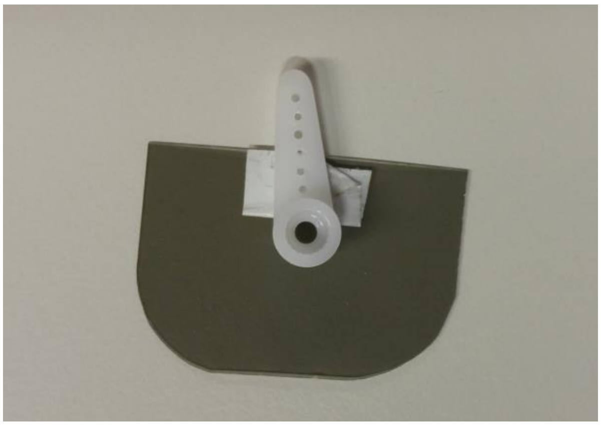
Next, we look for the biggest difference (`np.fmax`) and save it as the resulting image (`cv2.imwrite`). In the end, the servomotor is switched off again (`ServoOff`).

Step 6: Put it all together

If the polarizer is big enough for several Dolpi cams it might be advantageous to cut it into smaller portions so we may use it for multiple products. Think twice and be careful – polarizer film is rather expensive!

Glue the polarizer to the head part of the servomotor and stick the head part on the servomotor.





Switch off the Pi and disconnect it from power supply. Disconnect the HDMI adapter (including the monitor) – for future use, we will use the WIFI interface instead. Disconnect the USB hub. Instead, connect the webcam directly to the USB adapter. Now mount everything in the housing at its designated site and make sure the camera looks directly through the polarizer to the outside world.

Step 7: Try it out

Re-connect the Pi to the power supply, wait two or three minutes and try to connect to the WiFi network using a smartphone or computer. Does it still work?

Now, take several pictures. If the camera doesn't look through the polarizer film at all angles, you should re-adjust the head part of the servomotor.

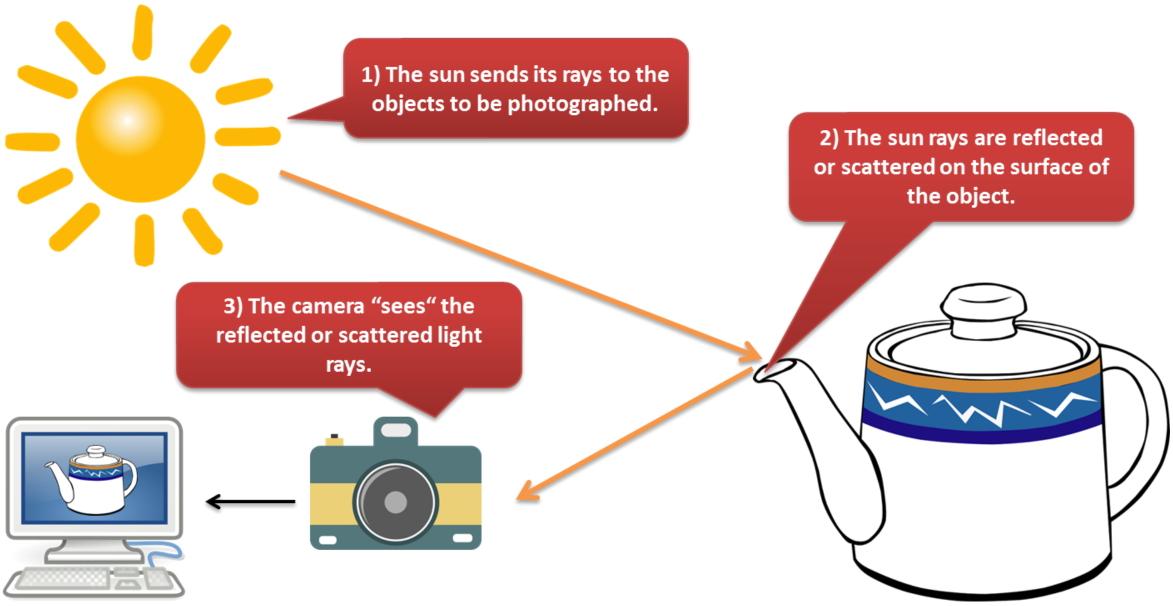
It is important not to move the camera during the measurement process as it will take three pictures in total and these pictures should be taken exactly from the same viewpoint.

If all works as desired, you should be able to tell apart linearly polarized light from unpolarized light. It is easiest to photograph an LCD screen. As you know, LCDs emit linearly polarized light. In the resulting picture, the screen should be shown in bright colours whereas the surroundings are rather black.

Concluding thoughts

You may ask what a Dolpi camera is good for, apart from identifying an LCD display (which is really no big deal). To answer this question, think back to what we learned about the working principle of ordinary cameras.

Cameras take light from the outer world. Where does this light come from? Well, unless you look directly into the sun or another light source, most light that is taken by the camera is somehow scattered or reflected on the surfaces of the objects that surround us, thereby making these objects visible:


1) The sun sends its rays to the objects to be photographed.
2) The sun rays are reflected or scattered on the surface of the object.
3) The camera "sees" the reflected or scattered light rays.

Light from the sun is completely unpolarized, that is, all different planes of polarisation are present. Normally, if light falls onto a surface, not all light is reflected. There may be also some light that is absorbed or even transmitted. The percentage of light that is reflected is called *reflectance*.

The reflectance is usually dependent on the wavelength (=colour) of the light, thereby giving objects their characteristic colour. Moreover, for some materials, the reflectance also exhibits a characteristic dependence on the plane of polarization – light will be linearly polarized once it is reflected on the surface.

And this is where our Dolpi cam comes in handy! With it, we can identify objects with a considerable difference of the reflectivity for different planes of polarisation. It turns out that many man-made materials (like glass, plastic, metal) belong to this group. So we may make them visible.

Some researchers claim that it is possible to detect landmines with Dolpi cameras as their surfaces are man-made and are therefore in contrast to the ground on which the lie. What do you think? Which challenges or problems might occur on the search for landmines using a Dolpi camera? Which chances do you see for this technology?

PHABLABS 4.0

PHABLABS 4.0 is a European project where two major trends are combined into one powerful and ambitious innovation pathway for digitization of European industry:

On the one hand the growing awareness of photonics as an important innovation driver and a key enabling technology towards a better society, and on the other hand the exploding network of vibrant Fab Labs where next-generation practical skills-based learning using KETs is core but where photonics is currently lacking.

www.PHABLABS.eu

This workshop was set up by the UoAS Wildau and the Ferdinand-Braun-Institut Berlin in close collaboration with ViNN:Lab at UoAS Wildau.

Technische Hochschule Wildau Technical University of Applied Sciences WILDAU

ViNN:Lab Think:Make:Share

FBH Leibniz Ferdinand Braun Institut

Imprint / Contact

Jacob Adolf - Florian Andert - Markus Lahr
Technical University of Applied Sciences Wildau
Hochschulring 1
15745 Wildau
Germany

Office: Building 16A, room 2095
ViNN:Lab: Building 16A, room 2095

E-mail: creativelab@th-wildau.de
Follow us on facebook: facebook.com/ViNNLab
Blog: vinnlab.de

PHOTONICS21

PHOTONICS PUBLIC PRIVATE PARTNERSHIP